Estimação de ângulos de orientação de uma IMU usando filtro de Kalman: Um estudo orientado ⋆

Igor Miranda Sartori * **e Fernando de Oliveira Souza** **

* *Programa de Pós-Graduação em Engenharia Elétrica - Universidade Federal de Minas Gerais - Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brasil (e-mail: igorms1307@ufmg.br)*
** *Departamento de Engenharia Eletrônica, Universidade Federal de Minas Gerais, MG (e-mail: fosouza@ufmg.br)*

Abstract: This paper presents a detailed study addressing the theory, application, and implementation of a Kalman filter for estimating orientation angles from an Inertial Measurement Unit (IMU). Experimental results obtained using an ESP32 microcontroller, an MPU-6050 sensor, and a breadboard validate the proposed approach. The integration of accelerometer and gyroscope measurements via the Kalman filter effectively compensates for these sensors' inherent noise and bias, resulting in enhanced measurement accuracy and reliability.

Resumo: Neste trabalho, é apresentado um estudo detalhado que aborda a teoria, aplicação e implementação de um filtro de Kalman para a estimação dos ângulos de orientação a partir de uma Unidade de Medida Inercial (IMU). Os resultados experimentais, obtidos com um microcontrolador ESP32, um sensor MPU-6050 e uma protoboard, validam a abordagem proposta. A integração das medições do acelerômetro e do giroscópio por meio do filtro de Kalman compensa eficazmente o ruído e o bias inerentes a esses sensores, proporcionando maior precisão e confiabilidade nas medições.

Keywords: Kalman Filter; Inertial Measurement Unit; Sensor Fusion; ESP32; MPU-6050.
Palavras-chaves: Filtro de Kalman; IMU; Fusão de Sensores; ESP32; MPU-6050.

1. INTRODUÇÃO

O filtro de Kalman é um estimador recursivo amplamente utilizado para estimar estados em sistemas dinâmicos, mesmo na presença de ruídos de processo e medição (Aguirre, 2015). Sua relevância está na capacidade de combinar, de forma inteligente, informações de diferentes fontes, proporcionando estimativas mais precisas e robustas. A obra clássica de Welch and Bishop (1995) oferece uma introdução prática e detalhada ao filtro de Kalman, abordando tanto o filtro discreto quanto suas variantes. Esse trabalho é amplamente reconhecido como uma referência essencial para o entendimento dos princípios do processamento recursivo de sinais, servindo de base para aplicações em áreas como navegação autônoma e fusão de dados de sensores.

Uma Unidade de Medida Inercial (IMU, do inglês *Inertial Measurement Unit*) integra sensores como acelerômetros e giroscópios para determinar a orientação e a posição de um objeto. No entanto, os acelerômetros são sensíveis a ruídos e interferências externas, enquanto as estimativas de ângulos obtidas por giroscópios sofrem com desvios acumulados devido à integração numérica dos sinais. Nesse cenário, a implementação de técnicas de fusão de sensores tem ganhado destaque para mitigar os problemas inerentes à medição isolada de cada sensor. O filtro de Kalman se destaca como uma solução eficaz, pois permite a fusão dos dados de ambos os sensores, compensando as limitações individuais de cada um (Lu et al., 2022). Em dos Santos Oliveira and Gonçalves (2017) é proposta a implementação do filtro de Kalman, que integra dados de acelerômetros e giroscópios para determinar ângulos de orientação.

Neste diapasão, o presente trabalho apresenta um estudo orientado que aborda a teoria, a aplicação e a implementação de um filtro de Kalman para estimar os ângulos de orientação de uma IMU. Para validar os conceitos teóricos, foram realizados experimentos práticos utilizando um microcontrolador ESP32, uma IMU (MPU-6050) e uma protoboard, conforme ilustrado na Figura 1 e detalhado na Tabela 1 as conexões elétricas.

Tabela 1. Conexão entre ESP32 e MPU6050

Pinos ESP32	Pinos MPU6050
3V	VCC
GND	GND
D22	SCL
D21	SDA

O objetivo central é integrar teoria e prática, equilibrando o conhecimento acadêmico com a aplicação real. Essa abordagem não apenas consolida os fundamentos teóricos, mas também demonstra a eficácia do filtro de Kalman em melhorar a precisão das medições de orientação, tornando-

⋆ O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001 e do Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil (CNPq)

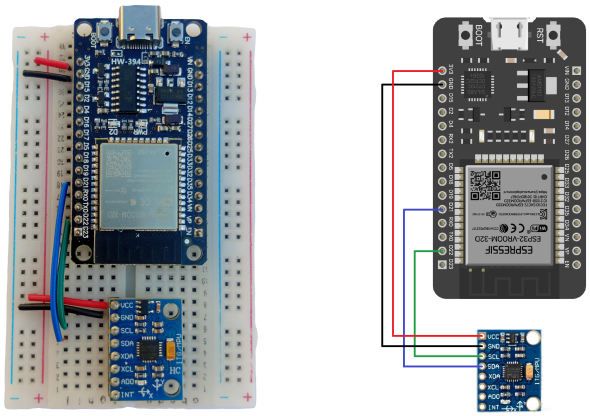

Figura 1. Conexão entre ESP32 e Módulo GY-521 MPU-605, montagem e esquemático, respectivamente.

o uma ferramenta essencial para o desenvolvimento de soluções avançadas em engenharia.

2. UNIDADE DE MEDIÇÃO INERCIAL (IMU)

Quando se utiliza uma IMU para determinar a orientação de um objeto, por meio dos ângulos de arfagem (pitch), rolagem (roll) e guianda (yaw), nota-se que tanto o acelerômetro quanto o giroscópio apresentam deficiências para realização desta tarefa.

O acelerômetro possui ruído elevado nas leituras, principalmente quando está sujeito a vibrações. O giroscópio apresenta leituras menos sensíveis à ruído, mas, devido à integração numérica para o cálculo do ângulo, um pequeno erro de medição (bias) pode gerar um desvio do valor real que cresce ao longo do tempo. Essa taxa de desvio é conhecida como bias.

Para ilustrarmos essas deficiências precisamos entender como cada um desses sensores funcionam separadamente. A seguir, para determinar o posicionamento do dispositivo usamos ϕ, θ e ψ, para denotar os ângulos roll, pitch e yaw, respectivamente.

2.1 Giroscópio

O giroscópio 3-eixos utiliza três giroscópios devidamente posicionados em cada eixo do sistema cartesiano tridimensional de forma a medir a velocidade angular em torno desses eixos (x, y, z) e, assim, utilizando integração numérica ao longo do tempo é possível estimar os ângulos de inclinação (*roll*, *pitch* e *yaw*) para determinar a orientação de um corpo.

Para ilustrar como determinar o posicionamento do dispositivo são definidas as velocidades angulares $\dot{\phi}$ como velocidade angular em torno do eixo-x do giroscópio, $\dot{\theta}$ como velocidade angular em torno do eixo-y do giroscópio e $\dot{\psi}$ como velocidade angular em torno do eixo-z do giroscópio. Logo, a estimativa do deslocamento angular pode ser obtida via integração numérica, por exemplo, usando $\dot{\phi}$, temos

$$\begin{aligned}\phi_k &= \sum_{i=0}^{k}(t_i - t_{i-1})\dot{\phi}_i \\ &= (t_k - t_{k-1})\dot{\phi}_k + \sum_{i=0}^{k-1}(t_i - t_{i-1})\dot{\phi}_i \\ &= (t_k - t_{k-1})\dot{\phi}_k + \phi_{k-1}, \end{aligned} \tag{1}$$

na qual ϕ_k denota o deslocamento angular calculado na iteração atual, ϕ_{k-1} denota o deslocamento angular calculado na iteração anterior, $\dot{\phi}_k$ a velocidade angular medida na iteração atual e $t_k - t_{k-1}$ o intervalo de tempo entre as amostragens. Note que a velocidade angular por si só não permite determinar a posição angular, mas apenas o deslocamento angular, pois não se conhece a condição inicial da posição angular.

2.2 Acelerômetro

O acelerômetro de três eixos utiliza três sensores devidamente posicionados para medir a aceleração específica atuando sobre um corpo no sistema cartesiano tridimensional. Em condições estáticas ou de baixa dinâmica, a principal componente dessa aceleração é a gravidade. Dessa forma, a partir das projeções do vetor gravidade nos eixos do sensor e do uso de relações trigonométricas, é possível estimar a orientação do corpo no espaço tridimensional.

Na Figura 2 é ilustrado o posicionamento dos acelerômetros em uma IMU considerando a orientação de referência típica por meio dos eixos x e y no plano horizontal e o eixo-z perpendicular ao plano horizontal. Note que os eixos do plano horizontal (xy) estão perpendiculares ao vetor gravidade e o vetor vertical z na mesma direção. Portanto, como os vetores x e y são ortogonais ao vetor gravidade g, nesta orientação do dispositivo a gravidade efetiva medida pelos acelerômetros nas direções x e y é nula. Por outro lado, a aceleração efetiva medida pelo acelerômetro na direção z é a aceleração da gravidade $1g$. Além disso, note que ao girar o sistema cartesiano tridimensional em torno do eixo z (*yaw*), a direção da gravidade nos eixos x e y não muda e, portanto, não é possível determinar o ângulo ψ usando apenas um acelerômetro.

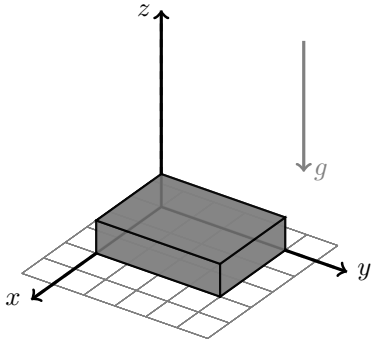

Figura 2. Ilustração do posicionamento e da orientação dos acelerômetros, em que cada sensor está alinhado com um dos eixos do sistema cartesiano tridimensional.

Na Figura 3 é ilustrado uma rotação no eixo-x, na qual pode-se notar que a projeção do vetor gravidade no eixo-y produz uma aceleração efetiva no eixo-y (Ay), medida

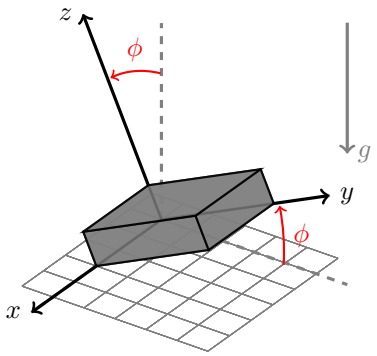

Figura 3. Ilustração do posicionamento do acelerômetro no plano cartesiano tridimensional sujeito a uma rotação ϕ no eixo-x.

pelo acelerômetro, igual ao seno do ângulo entre o eixo-y do acelerômetro e o plano horizontal, ou seja, $Ay = g\,\text{sen}(\phi)$. Por outro lado, uma projeção do vetor gravidade no eixo-z produz uma aceleração efetiva (Az), medida pelo acelerômetro, igual ao cosseno do ângulo entre o plano vertical e o eixo-z do acelerômetro, ou seja, $Az = g\cos(\phi)$. Portanto, combinando as duas equações anteriores, temos

$$\phi = \arctan\left(\frac{Ay}{Az}\right).$$

Seguindo uma metodologia similar, pode-se obter uma equação para determinar ϕ a partir de uma orientação do dispositivo na qual todos os seus eixos estão deslocados dos eixos do plano cartesiano de referência, conforme ilustrado na Figura 4, onde são apresentados apenas os eixos do dispositivo. Note que, neste caso, pode ser considerado o vetor resultante r_{xz} da soma dos eixos x e z; assim, a aceleração efetiva no vetor resultante é igual ao cosseno do ângulo entre este vetor resultante e o plano vertical, ou seja, $Ar_{xz} = g\cos(\phi)$. Entretanto, a aceleração efetiva Ar_{xz} não é medida pelo acelerômetro, mas pode ser expressa em termos das acelerações efetivas medidas nos eixos x e y pela equação $Ar_{xz} = \sqrt{Ax^2 + Az^2}$. Portanto, combinando estas últimas equações com $Ay = g\,\text{sen}(\phi)$, resulta em

$$\phi = \arctan\left(\frac{Ay}{\sqrt{Ax^2 + Az^2}}\right).$$

De forma similar, é possível determinar o ângulo θ. Note que, ao rotacionar os eixos do dispositivo e trocar ϕ por θ na Figura 4, chega-se a seguinte equação:

$$\theta = \arctan\left(\frac{-Ax}{\sqrt{Ay^2 + Az^2}}\right).$$

Para garantir que o ângulo de θ seja calculado em conformidade com a convenção de orientação do sistema de referência (positivo para inclinação para frente e negativo para inclinação para trás), foi adicionado um sinal negativo à expressão.

2.3 Experimentos para estimação de ângulo

Para ilustrar as deficiências mencionadas anteriormente nas estimativas dos valores dos ângulos obtidas pelo giroscópio (desvio) e acelerômetro (ruído), foram realizadas medições utilizando o sensor MPU-6050. Durante os experimentos, o sensor permaneceu em repouso, e o ângulo ϕ foi

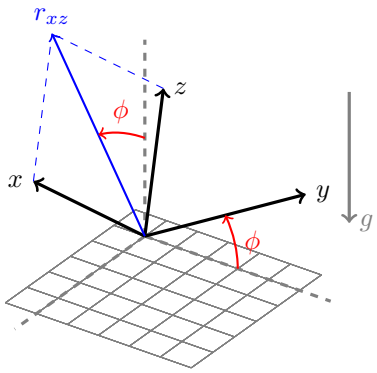

Figura 4. Ilustração do ângulo ϕ considerando um posicionamento do dispositivo no qual seus eixos estão deslocados dos eixos do plano cartesiano de referência.

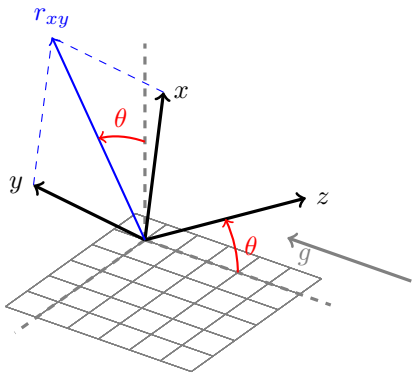

Figura 5. Ilustração do ângulo θ considerando um plano cartesiano de referência no qual o vetor gravidade é perpendicular ao plano xy.

estimado de duas formas: uma utilizando os dados do giroscópio e outra, os do acelerômetro. Os resultados podem ser visualizados nas Figuras 6(a) e 6(c), respectivamente.

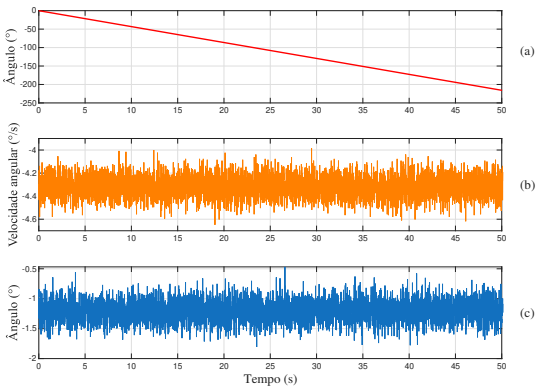

Figura 6. (a) Ângulo ϕ calculado ao longo do tempo utilizando o giroscópio do eixo x. (b) Velocidade angular medida $\dot{\phi}$ ao longo do tempo utilizando o giroscópio do eixo x. (c) Ângulo ϕ calculado ao longo do tempo utilizando os acelerômetros dos eixos y e z.

Observa-se que o ruído presente nas estimativas do acelerômetro é bastante elevado, enquanto o *bias* compromete significativamente a estimação do ângulo realizada pelo giroscópio ao longo do tempo. Dessa forma, uma estratégia eficaz no uso da IMU é combinar as leituras dos

dois sensores, suavizando as características indesejáveis de cada um. Essa abordagem, conhecida como fusão de sensores, foi implementada neste trabalho por meio do Filtro de Kalman, que permite tanto a estimação do *bias* quanto a rejeição do ruído, proporcionando medições de maior qualidade em comparação às obtidas por cada sensor isoladamente.

3. IMPLEMENTAÇÃO DO FILTRO DE KALMAN

Nesta seção vamos apresentar os componentes matemáticos, as etapas envolvidas no filtro de Kalman e como implementá-lo em linguagem C++ utilizando a plataforma Arduino. Para mais detalhes sobre o filtro de Kalman, veja Aguirre (2015).

3.1 Componentes Necessários

Para a implementação do filtro de Kalman clássico é necessário:

- Modelo do sistema: um modelo linear e invariante no tempo em espaço de estados que descreva a dinâmica do sistema na forma
$$x_k = Ax_{k-1} + Bu_k + Dw_k \tag{2}$$
$$y_k = Cx_k + v_k \tag{3}$$
sendo x_k o vetor de estado do sistema no instante k, u_k o sinal de entrada, y_k a saída medida, w_k o ruído do processo e v_k o ruído de medição.
- Matrizes de covariância: do ruído do processo (Q), que representa a incerteza no modelo, e do ruído das medições (R), que descreve a incerteza nas medições.
- Estado inicial: estimativa inicial do estado do sistema (x_0) e covariância inicial ($P_{0,0}$), que denota a incerteza inicial na estimativa do estado.

Para determinar o modelo do sistema, conforme discutido na Seção 2.1, podemos utilizar a medida do giroscópio para estimar o ângulo, de forma similar ao modelo de predição em (2). Ademais, conforme ilustrado anteriormente na Figura 6(a), é sabido que a medida do giroscópio contém um *bias* de medição, o qual é desconhecido *a priori*, ou seja, o valor da medida do giroscópio é diferente do valor real e o valor desse desvio é o *bias*. Assim, o valor do ângulo (por exemplo, ϕ) pode ser calculado aproximadamente a partir da medida do giroscópio da seguinte forma

$$\begin{aligned}\phi_k &= \phi_{k-1} + T_s(\dot{\phi}_{g,k} - b_{k-1} + w_{g,k}) \\ &= \phi_{k-1} + T_s(\dot{\phi}_{g,k} - b_{k-1}) + T_s w_{g,k},\end{aligned} \tag{4}$$

sendo que ϕ_k representa o valor do ângulo na iteração atual, ϕ_{k-1} o valor do ângulo na iteração anterior, T_s o tempo de amostragem (assumido constante), $\dot{\phi}_{g,k}$ representa a medida do giroscópio, b_{k-1} o valor do *bias* na medida do giroscópio e $w_{g,k}$ o ruído na medida do giroscópio. Basicamente a equação acima aproxima uma operação de integração numérica ao longo do tempo e, comparando (4) com a equação do modelo em (2) notamos que a medida do giroscópio é a entrada do modelo.

Além disso, assumimos que

$$b_k = b_{k-1} + w_{b,k}, \tag{5}$$

na qual $w_{b,k}$ representa o ruído no *bias* do giroscópio. Por fim, a partir da medida do acelerômetro obtemos ϕ_a, o valor do ângulo adicionado de um ruído de medição, ou seja

$$\phi_{a,k} = \phi_k + v_k, \tag{6}$$

a qual pode ser representada pela equação (3), correspondente à equação de medição do modelo considerado no filtro de Kalman. Dessa forma, a medida do acelerômetro constitui a saída do modelo.

Portanto, um modelo que descreve como o ângulo e o *bias* evoluem ao longo do tempo a partir das medições da IMU pode ser representado pelo modelo em (2)-(3), fazendo

$$x_k = \begin{bmatrix}\phi \\ b\end{bmatrix}_k, \quad A = \begin{bmatrix}1 & -T_s \\ 0 & 1\end{bmatrix}, \quad B = \begin{bmatrix}T_s \\ 0\end{bmatrix},$$

$$C = [1\ 0], \quad D = \begin{bmatrix}T_s & 0 \\ 0 & 1\end{bmatrix}, \quad w_k = \begin{bmatrix}w_g \\ w_b\end{bmatrix}_k, \quad y_k = \phi_{a,k}.$$

Ademais, para determinar as matrizes de covariância, é assumido que apenas ruídos no processo atuam em cada um dos estados, ângulo e *bias*, de forma independente, ou seja, a matriz de covariância do ruído de processo é composta apenas pela variância do ruído em cada estado,

$$Q_k = \begin{bmatrix}Q_g & 0 \\ 0 & Q_b\end{bmatrix}.$$

Neste caso, a matriz de covariância do ruído de medição é apenas um escalar (assumido constante), que representa a variância do ruído de medição, assumida constante:

$$R_k = R_{k-1}.$$

Para determinar os parâmetros Q_g e R, foram calculadas as variâncias de dados experimentais do giroscópio e do acelerômetro, ilustrados nas Figuras 6(b) e 6(c), resultando em $Q_g = 0{,}0088$ e $R = 0{,}0305$. Como a variância do *bias* não pode ser medida diretamente, escolhemos $Q_b = Q_g$. É importante destacar que o ruído de processo no modelo (4) não se restringe apenas ao ruído intrínseco dos sensores, como o modelo obtido aqui a partir das equações (1), (5) e (6), mas também abrange incertezas no modelo dinâmico, perturbações externas, erros de discretização e outros fatores que impactam a precisão da previsão do estado. Dessa forma, determinar as matrizes de covariância com exatidão pode não ser viável, e métodos empíricos são frequentemente utilizados para ajustar os valores dessas matrizes. Neste trabalho, não realizaremos ajustes adicionais nessas matrizes, embora tal procedimento pudesse contribuir para aumentar a eficiência do filtro de Kalman projetado.

Por fim, por não conhecer o estado inicial assumimos condições iniciais nulas e que ϕ e b não estão correlacionados,

$$x_0 = \begin{bmatrix}\phi \\ b\end{bmatrix}_0 = \begin{bmatrix}0 \\ 0\end{bmatrix} \quad \text{e} \quad P_{0,0} = \begin{bmatrix}1 & 0 \\ 0 & 1\end{bmatrix}.$$

Com as informações necessárias em mãos, estamos prontos para implementar o filtro de Kalman, conforme detalhado na próxima seção.

3.2 Operação do Filtro de Kalman

De forma sucinta, na presente aplicação, o filtro de Kalman estima o vetor de estados $\hat{x}_{k|k}$ usando o modelo do sistema e sua entrada, medida do giroscópio, e da saída medida do processo, y_k, obtida pelo acelerômetro, da seguinte forma:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\left(y_k - C\hat{x}_{k|k-1}\right) \tag{7}$$

Nota-se que a ideia básica consiste em corrigir a estimativa do estado por meio das medições dos sensores, ponderadas pela matriz K_k, conhecida como ganho de Kalman.

Como mostrado, o filtro de Kalman é um algoritmo no qual um conjunto de equações devem ser computadas de forma sequencial. Este algoritmo é geralmente dividido em duas etapas: etapa de predição e etapa de correção.

A etapa de predição é realizada sem utilizar informações atuais. Ela é composta pelo cômputo de duas equações, cujo objetivo é prever o estado atual e a matriz de covariância do erro. A seguinte equação é utilizada para prever o estado atual

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_{k-1}, \tag{8}$$

a qual fornece o estado do processo baseando-se no modelo, sendo $\hat{x}_{k|k-1}$ o estado estimado no instante k utilizando informações anteriores e $\hat{x}_{k-1|k-1}$ o estado estimado na iteração anterior.

A próxima equação é utilizada para estimar a matriz de covariância do erro

$$P_{k|k-1} = AP_{k-1|k-1}A^T + DQ_{k-1}D^T, \tag{9}$$

sendo $P_{k|k-1}$ a matriz de covariância de $\hat{x}_{k|k-1}$ no instante k utilizando informações anteriores e $P_{k-1|k-1}$ a matriz de covariância de $\hat{x}_{k-1|k-1}$. Note que $P_{k|k-1}$ estima a confiança nos valores atuais do estado estimado e, assim, a última equação ilustra que a confiança no estado estimado diminui devido à incerteza no modelo usado.

Na etapa de correção, utilizando informações atuais, é feita a correção das quantidades estimadas na etapa de predição resultando na estimativa definitiva do vetor de estados pelo filtro de Kalman. A primeira equação desta etapa é

$$K_k = P_{k|k-1}C^T[CP_{k|k-1}C^T + R_{k-1}]^{-1}, \tag{10}$$

na qual K_k é utilizado para realizar a correção dos valores estimados na etapa de predição. Em seguida, de posse de K_k, os valores estimados na etapa de predição são corrigidos

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k[y_k - C\hat{x}_{k|k-1}] \tag{11}$$

e

$$P_{k|k} = P_{k|k-1} - K_kCP_{k|k-1}. \tag{12}$$

3.3 Algoritmo em C++

Para a implementação do filtro, é necessário inicializar os parâmetros do filtro, conforme estabelecido anteriormente.

```
float Q_gyro = 0.0088;
float Q_bias = 0.0088;
float R_meas = 0.0305;
float angle = 0.0;
float bias = 0.0;
float P[2][2] = {{ 1, 0 }, { 0, 1 }};
float K[2] = {0, 0};
```

No trecho anterior, `Q_gyro` $= Q_g$, `Q_bias` $= Q_b$, `R_meas` $= R_k$ e `P[2][2]` $= P_{0,0}$. Além disso, são definidas condições iniciais nulas para o vetor de estados do modelo: `angle` representa o ângulo estimado e `bias` corresponde ao *bias* de medição do giroscópio. Por fim, `K[2]` armazena o ganho de Kalman.

Conforme discutido na seção 3.2, é fácil notar que a primeira equação do filtro (8) nada mais é que o cálculo do valor do ângulo usando a medida do giroscópio (1), desprezando o ruído. Portanto, em linguagem C++, a seguinte linha implementa a primeira equação do filtro:

```
angle = angle + Ts*(gyro - bias);
```

Acima, `Ts` é o tempo de amostragem e `gyro` a medida do giroscópio.

Para o modelo considerado, a matriz $P_{k|k-1}$ e a matriz D em (9) possuem dimensão 2×2. Assim, a partir de (9), temos:

$$\begin{aligned}
&\begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k-1} \\
&= \begin{bmatrix} 1 & -T_s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k-1|k-1} \begin{bmatrix} 1 & 0 \\ -T_s & 1 \end{bmatrix} + \begin{bmatrix} T_s & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} Q_\phi & 0 \\ 0 & Q_b \end{bmatrix} \begin{bmatrix} T_s & 0 \\ 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} P_{00} - T_sP_{10} & P_{01} - T_sP_{11} \\ P_{10} & P_{11} \end{bmatrix}_{k-1|k-1} \begin{bmatrix} 1 & 0 \\ -T_s & 1 \end{bmatrix} + \begin{bmatrix} T_s^2Q_\phi & 0 \\ 0 & Q_b \end{bmatrix} \\
&= \begin{bmatrix} P_{00} - T_sP_{10} - T_s(P_{01} - T_sP_{11}) & P_{01} - T_sP_{11} \\ P_{10} - T_sP_{11} & P_{11} \end{bmatrix}_{k-1|k-1} \\
&\quad + \begin{bmatrix} T_s^2Q_\phi & 0 \\ 0 & Q_b \end{bmatrix} \\
&= \begin{bmatrix} P_{00} - T_sP_{10} - T_s(P_{01} - T_sP_{11}) + T_s^2Q_\phi & P_{01} - T_sP_{11} \\ P_{10} - T_sP_{11} & P_{11} + Q_b \end{bmatrix}_{k-1|k-1} \\
&= \begin{bmatrix} P_{00} + T_s(T_sP_{11} - P_{01} - P_{10} + T_sQ_\phi) & P_{01} - T_sP_{11} \\ P_{10} - T_sP_{11} & P_{11} + Q_b \end{bmatrix}_{k-1|k-1},
\end{aligned}$$

a qual pode ser implementada em linguagem C++ da seguinte forma:

```
P[0][0] = P[0][0] + (P[1][1] * Ts - P[0][1]
          - P[1][0] + Q_gyro * Ts) * Ts;
P[0][1] = P[0][1] - P[1][1] * Ts;
P[1][0] = P[1][0] - P[1][1] * Ts;
P[1][1] = P[1][1] + Q_bias;
```

Agora, analisando a terceira equação no filtro (10), temos

$$\begin{aligned}
K_k &= P_{k|k-1}C^T[CP_{k|k-1}C^T + R_k]^{-1} \\
&= \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \left[\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + R_k \right]^{-1} \\
&= \begin{bmatrix} P_{00} \\ P_{10} \end{bmatrix}_{k|k-1} [P_{00} + R]^{-1}_{k|k-1},
\end{aligned}$$

que pode ser implementada da seguinte forma:

```
K[0] = P[0][0] / (P[0][0] + R_meas);
K[1] = P[1][0] / (P[0][0] + R_meas);
```

Em seguida, considerando a quarta equação do filtro: (11)

$$\begin{aligned}
\hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k[y_k - C\hat{x}_{k|k-1}] \\
\begin{bmatrix} \phi \\ b \end{bmatrix}_{k|k} &= \begin{bmatrix} \phi \\ b \end{bmatrix}_{k|k-1} + \begin{bmatrix} K_0 \\ K_1 \end{bmatrix}_k \left[y_k - \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ b \end{bmatrix}_{k|k-1} \right] \\
&= \begin{bmatrix} \phi \\ b \end{bmatrix}_{k|k-1} + \begin{bmatrix} K_0 \\ K_1 \end{bmatrix}_k [y_k - \phi_{k|k-1}],
\end{aligned}$$

que pode ser implementada da seguinte forma:

```
angle = angle + K[0] * (accel - angle);
bias  = bias + K[1] * (accel - angle);
```

Acima, `accel` $= y_k$ é a saída do processo medida pelo acelerômetro.

Finalmente, analisando a última equação do filtro (12), temos:

$$\begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k} = \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k-1} - \begin{bmatrix} K_0 \\ K_1 \end{bmatrix}_k \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k-1}$$

$$= \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix}_{k|k-1} - \begin{bmatrix} K_{0_k} P_{00_{k|k-1}} & K_{0_k} P_{01_{k|k-1}} \\ K_{1_k} P_{00_{k|k-1}} & K_{1_k} P_{01_{k|k-1}} \end{bmatrix},$$

que em linguagem C++ pode ser implementada da seguinte forma:

```
float P00_temp = P[0][0];
float P01_temp = P[0][1];
P[0][0] = P[0][0] - K[0] * P00_temp;
P[0][1] = P[0][1] - K[0] * P01_temp;
P[1][0] = P[1][0] - K[1] * P00_temp;
P[1][1] = P[1][1] - K[1] * P01_temp;
```

4. EXPERIMENTOS

Para ilustrar os resultados obtidos com o filtro de Kalman, implementado conforme o procedimento descrito na seção anterior, foi repetido o mesmo experimento apresentado na Seção 2.3, no qual a MPU-6050 foi mantida imóvel e o ângulo ϕ foi estimado, agora utilizando o filtro de Kalman. Os resultados são ilustrados na Figura 7, que mostra as estimativas do ângulo ϕ obtidas pelo filtro de Kalman e as medições individuais do acelerômetro e do giroscópio.

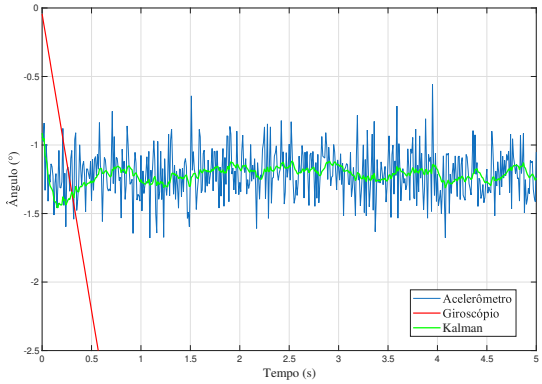

Figura 7. Estimativas do ângulo ϕ obtidas pelo filtro de Kalman, sobrepostas às medições individuais do acelerômetro e do giroscópio.

Além disso, com o objetivo de avaliar o desempenho do filtro em transições abruptas do ângulo ϕ, realizou-se um experimento no qual foi aplicada manualmente uma variação brusca de aproximadamente -180°, partindo de 90°. Os resultados desse ensaio são apresentados na Figura 8. Observa-se que o filtro mantém um comportamento consistente mesmo diante de variações extremas, demonstrando robustez na estimação do ângulo ϕ. Mais detalhes sobre os experimentos e códigos utilizados podem ser encontrados em Sartori (2025).

5. CONCLUSÃO

Este trabalho teve como objetivo principal evidenciar, de forma didática, a importância da integração de múltiplos sensores como estratégia para mitigar as limitações individuais, promovendo uma abordagem mais robusta e eficiente para a estimativa de estados. Para isso, foi desenvolvido um estudo dirigido que apresentou a implementação

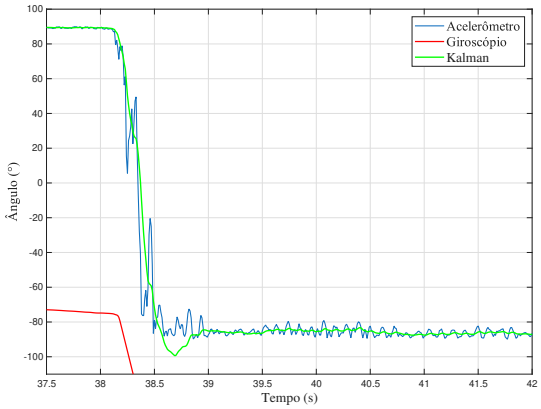

Figura 8. Estimativas do ângulo ϕ obtidas pelo filtro de Kalman, sobrepostas às medições individuais do acelerômetro e do giroscópio após variar o angulo de 90° a -90°.

de um filtro de Kalman aplicado à estimativa dos ângulos de orientação de uma IMU.

O presente trabalho demonstrou as vantagens de utilizar o filtro de Kalman para a fusão dos dados provenientes do acelerômetro e do giroscópio, com foco na estimativa precisa do ângulo de rolagem (roll). Por meio de experimentos realizados com o microcontrolador ESP32 e o sensor MPU-6050, foi possível evidenciar que, enquanto o acelerômetro sofre com elevados níveis de ruído e o giroscópio acumula desvios devido à integração numérica, a combinação dos sinais por meio do filtro de Kalman permite compensar essas deficiências. Essa abordagem resultou em medições mais confiáveis, validando a eficácia do método.

Em síntese, este estudo reforça a relevância do uso do filtro de Kalmam como uma técnica de fusão de sensores em aplicações que demandam precisão e confiabilidade na estimativa de estados.

6. MATERIAL SUPLEMENTAR

O código-fonte da implementação do filtro de Kalman, além de outras informações, estão disponíveis no repositório:

`https://github.com/SartoriDynamics/IMU`

REFERÊNCIAS

Aguirre, L.A. (2015). *Introdução à Identificação de Sistemas*. Editora UFMG, Belo Horizonte, 4a Edição edition.

dos Santos Oliveira, W. and Gonçalves, E.N. (2017). Implementação em C: filtro de Kalman, fusão de sensores para determinação de ângulos. *ForScience*, 5(3).

Lu, J., Ye, L., Zhang, J., Luo, W., and Liu, H. (2022). A new calibration method of mems imu plus fog imu. *IEEE Sensors Journal*, 22(9), 8728–8737.

Sartori, I.M. (2025). Repositório IMU. `https://github.com/SartoriDynamics/IMU`. Acesso em: 13 nov. 2025.

Welch, G. and Bishop, G. (1995). An introduction to the Kalman filter. Technical report, USA.